



Fachseminar



Thema heute:

Mobile Computing

*Beschreibung und Lokalisierung von
Diensten in drahtlosen Netzen*

Leanza Antonino



Uebersicht

- **Einleitung & Motivation**
- Service-Lokalisierung und Beschreibung in...
 - Bluetooth (SDP)
 - IrDA (IAS, IAP)
 - SLP
 - Jini
 - UPnP
 - Salutation
- Vergleich



Motivation

Die Wunschvorstellung der spontanen Vernetzung

■ „Keinerlei“ manuelle Konfiguration

- Gerätespezifische Parameter werden so gut es geht automatisch von den beteiligten Geräten ausgehandelt (“Selbstkonfigurierend”)

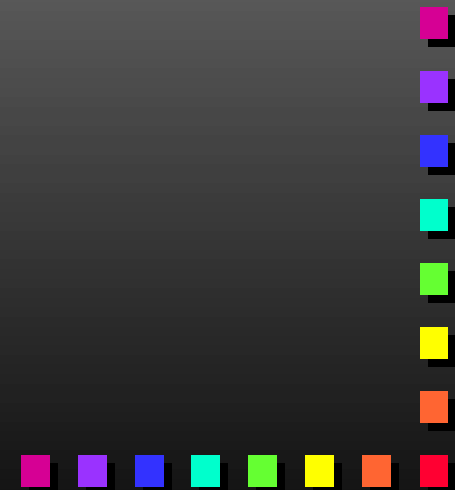
■ „Keinerlei“ manuelle Administration

- Möglichst keinen externen Wartungsaufwand
- Dynamische Verbindung von Dienst-Nutzer/Anbieter
- Einfache **Beschreibung, Lokalisierung und Nutzung von Diensten**

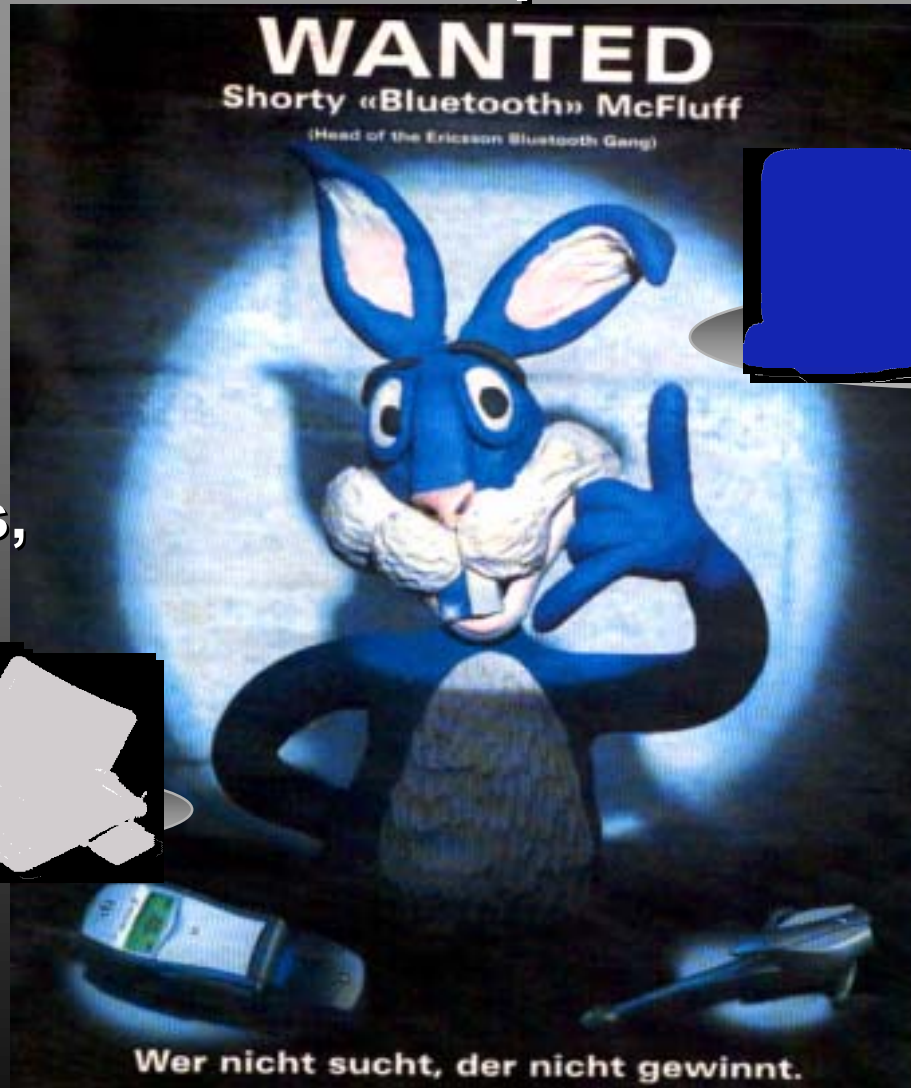


Spontane Vernetzung

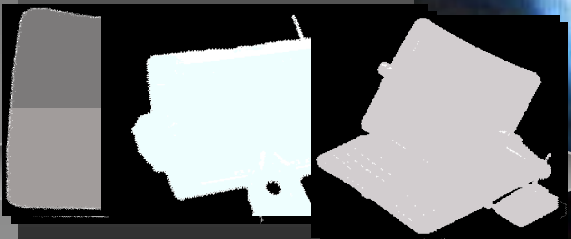
- Automatisch ablaufende Integration von Diensten in bestehenden Umgebungen
- Selbstkonfigurierend
- Sofortige Vernetzung und Nutzung von Geräten



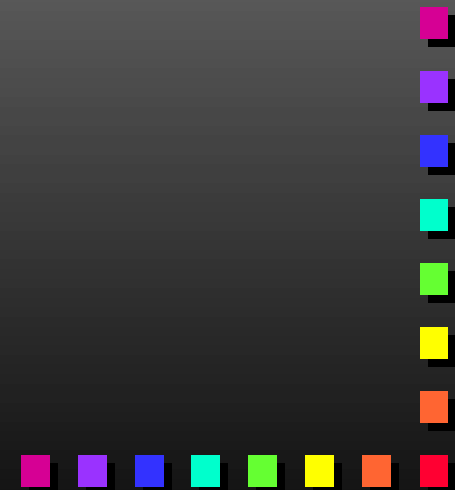
Beispiele



Books, tablets,
handheld PCs



TVs,
games



Umfeld / Infrastruktur

Dienst-Nutzung

(Service-Protokolle, i.d.R. unabhängig von D.-Vermittlung)

Dienst-Vermittlung

(Informationsmodelle, **Dienst-Beschreibung -Anfrage**)

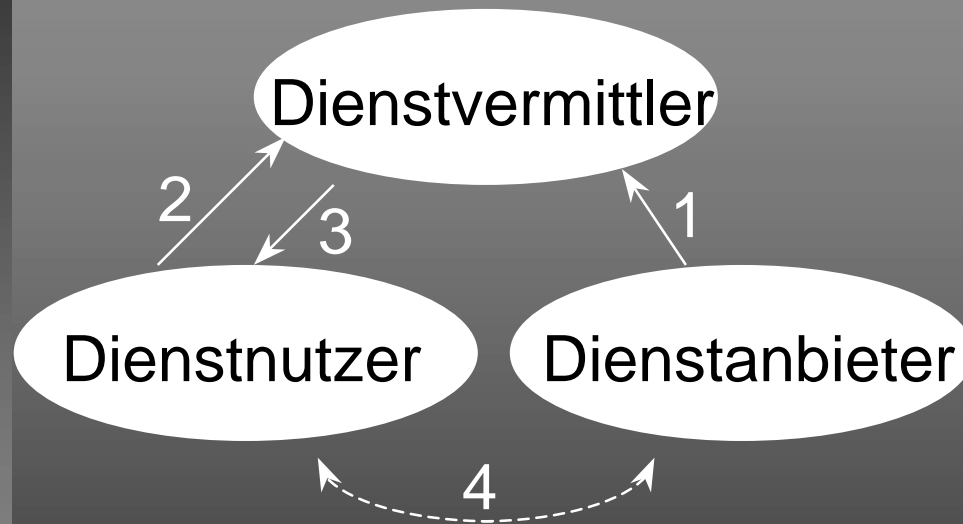


Netzwerk

(IP-Adressen, Kommunikations-Parameter, ...)

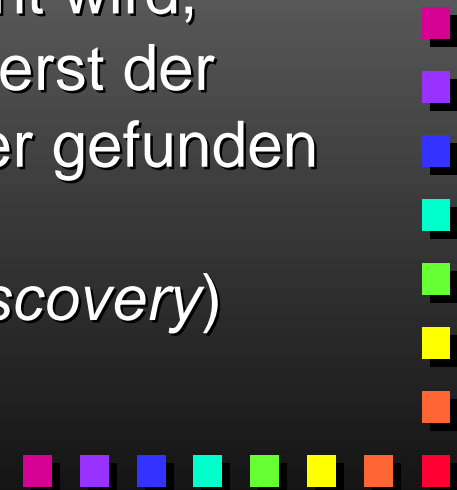


Genereller Ablauf der Dienstvermittlung



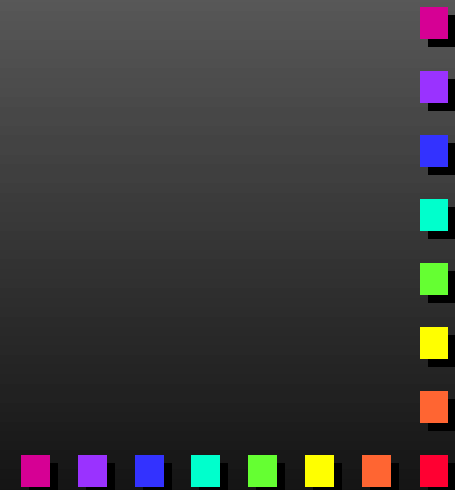
- 1: Registrierung
- 2: Dienstanfrage
- 3: Antwort
- 4: Dienstnutzung

- Bestimmte Implementierungen verzichten auf Dienstvermittler
- Falls Vermittlung gebraucht wird, muss zuerst der Vermittler gefunden werden (sog. *Discovery*)



Probleme

- ? Beschreibung
- ? Anfrage
- ? Benutzung
- ? Lokalisierung



Dienstbeschreibung [1/2]

- Open World Assumption
 - Dienste sind im Voraus nicht bekannt
(zur Entwicklungszeit des Klienten-Programmes)
 - Semantische Beschreibung der Dienste nötig
 - Im Voraus unbekannte Kombinationen ermöglicht
 - Vermitteln mit Hilfe von semantischen Informationen ist sehr schwierig und wird in der künstlichen Intelligenz behandelt



Dienstbeschreibung [2/2]

- Closed World Assumption
 - Struktur der Dienste ist im Voraus bekannt
 - Dienste sind mittels spezifizierter Schnittstellen beschrieben
 - Auswahl eines passenden Dienstes ist in diesem Fall einfacher: finde eine passende Schnittstelle unter der Menge der vorhandenen Schnittstellen
- Wir behandeln Systeme dieser Art



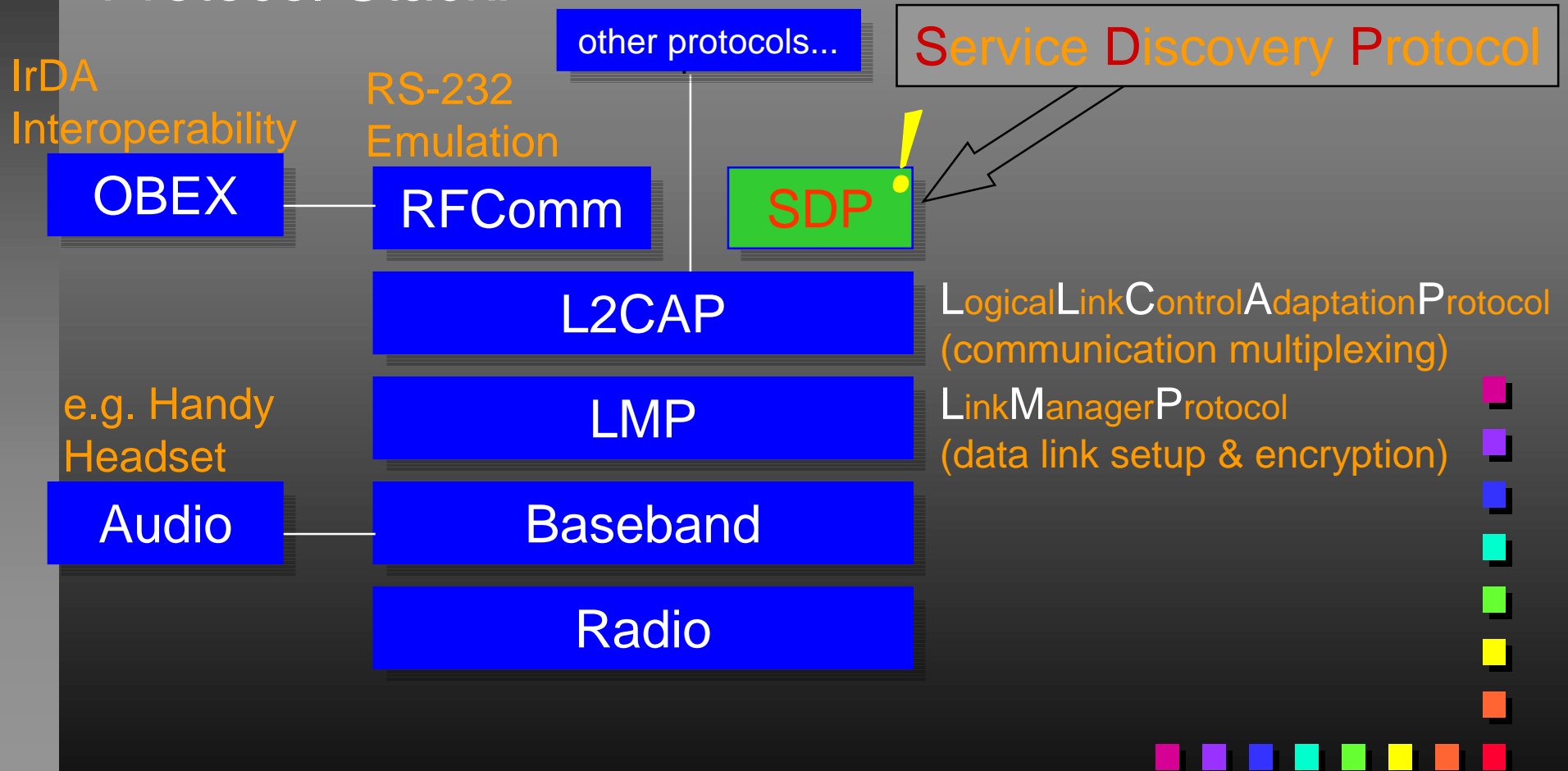
Uebersicht

- Einleitung & Motivation
- Service-Lokalisierung und Beschreibung in...
 - **Bluetooth (SDP)**
 - IrDA (IAS, IAP)
 - SLP
 - Jini
 - UPnP
 - Salutation
- Vergleich



Bluetooth

Protocol Stack:



Bluetooth / SDP

- Jedes von max. 8 Geräten im *Piconet* kann 1 SDP-Server implementieren
- Dienste eines Gerätes werden beim SDP-Server des jeweiligen Gerätes registriert
- Man kann SDP-Server nach bestimmten Diensten befragen



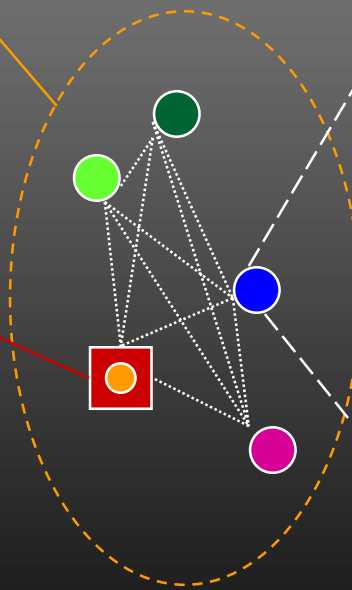
Bluetooth / SDP

Service-Beschreibung:

- Server verwaltet Menge von *Service Records*

Piconet

Master



Gerät das SDP-Server implementiert:

Menge von Service Records {

ID 32 bit;

Menge von ServiceAttributes {

AttributeID 16 bit;

AttributeValue Type;

}

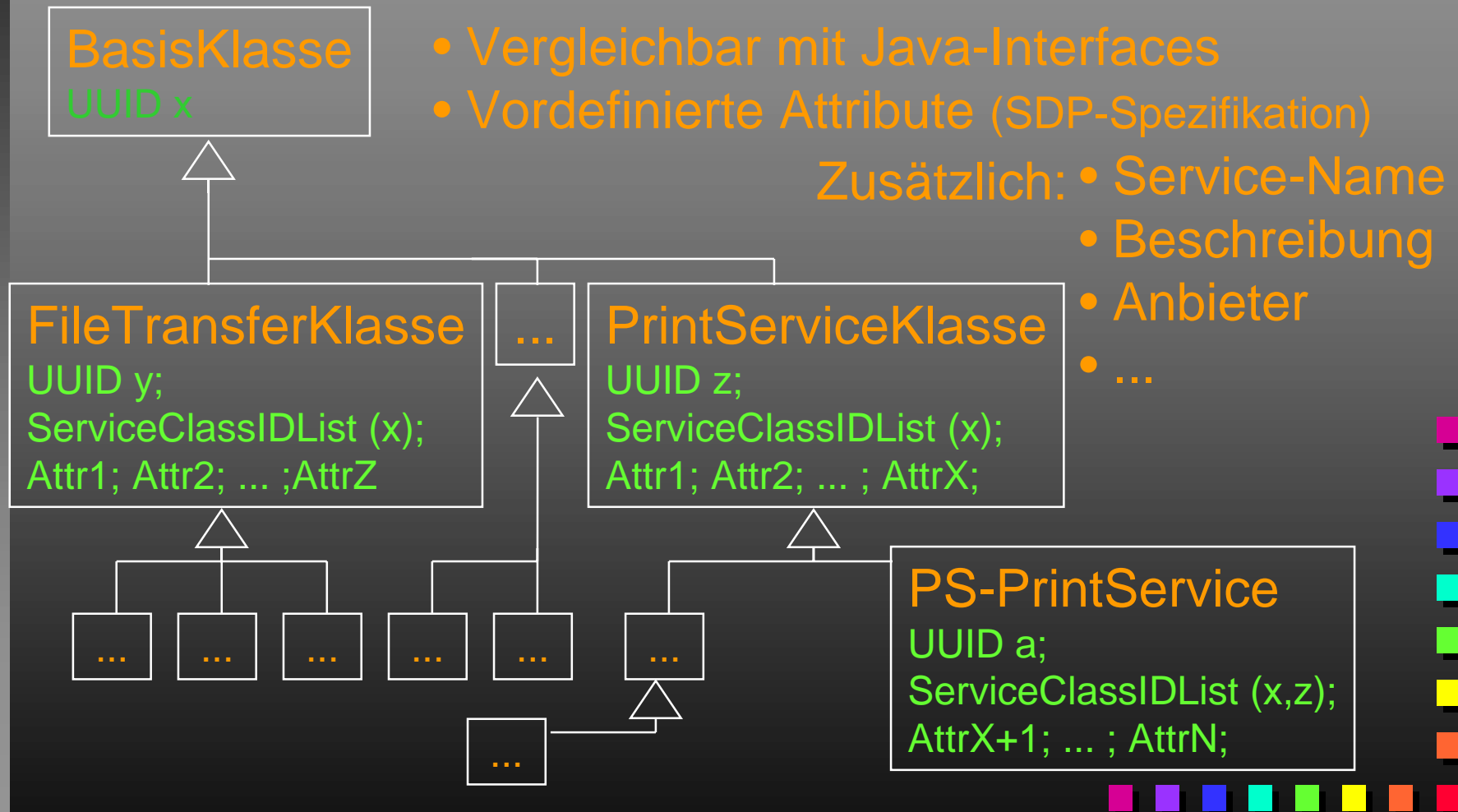
[...] ...

}

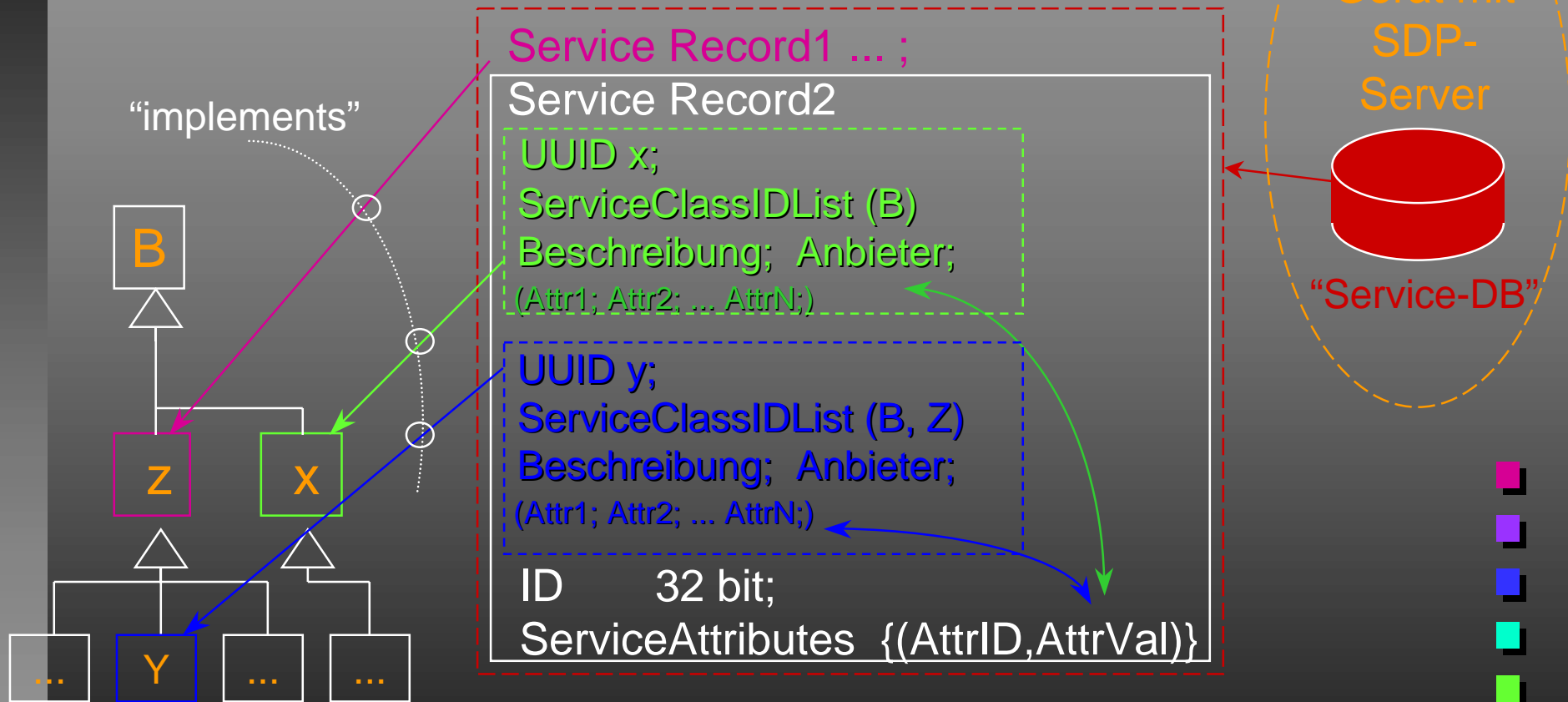


Bluetooth / SDP

Service-Klassen-Vererbungs-Hierarchie:



Bluetooth / SDP



Service-Record: Instanz einer Menge von Service-Klassen



Bluetooth / SDP

Service-Suchmechanismus

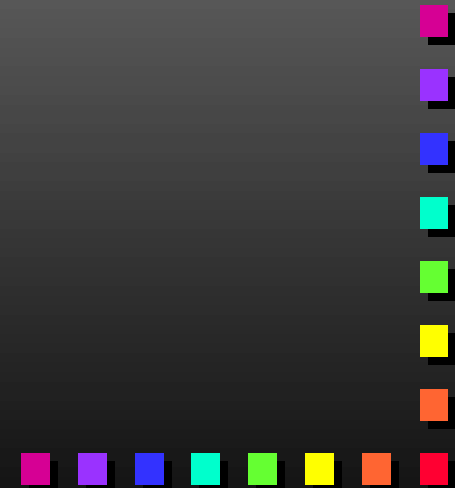
- Es kann *nur nach UUID's* gesucht werden
 - Typen wie Zahlen und Zeichenketten können nicht bei Suche verwendet werden
- Klient spezifiziert Suchmuster = *Liste von UUID's*
- Muster wird an den SDP-Server übertragen und mit jedem Service Record des Servers verglichen
- Resultat: Information, ob Menge der Service-klassenattribute des Gerätes eine Obermenge des Suchmusters darstellt



Bluetooth / SDP

Service-Suchmechanismus

- Keine zentrale “*Directory*”-Komponente
- Es kann nur nach UUID's von Klassen und Attributen gesucht werden kann
 - Dafür: Navigationsmöglichkeiten

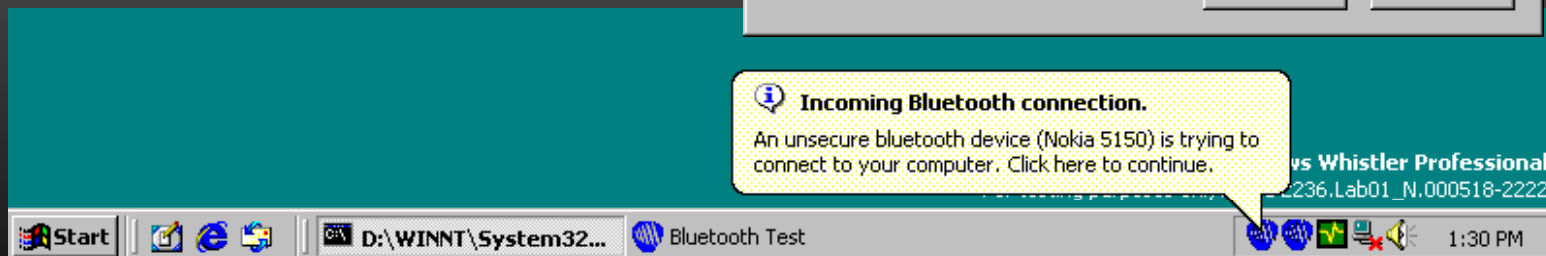
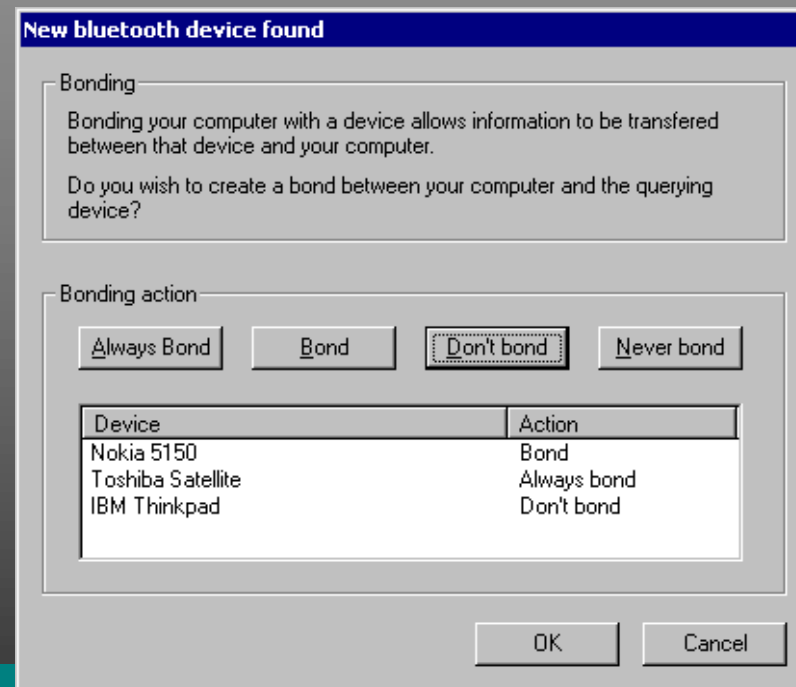


Bluetooth / SDP

Service-Suchmechanismus

Bsp.:

- Device Discovery unter *Windows*



Uebersicht

- Einleitung & Motivation
- Service-Lokalisierung und Beschreibung in...
 - Bluetooth (SDP)
 - IrDA (IAS, IAP)
 - SLP
 - Jini
 - UPnP
 - Salutation
- Vergleich

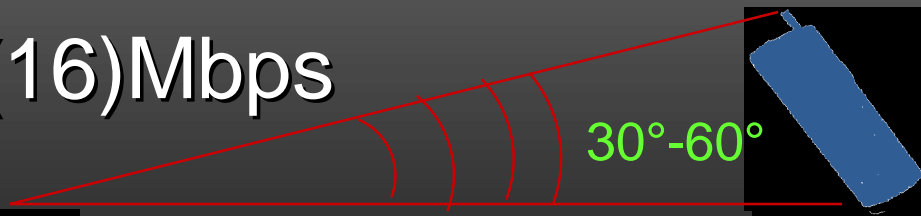


IrDA

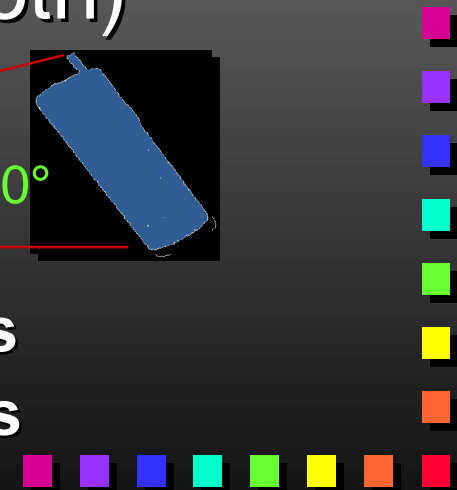


Desktops,
Notebooks

- Verbreitung um 1994
 - >40 Mio. existierende IrDA-Geräte
 - billig
- “Kabelersatz” (wie bei Bluetooth)
- bis 1m, 4 (16)Mbps



Phones, Pagers
PC companions



IrDA

Protokoll-Stack:

Opt.Layers

...

Tiny TP

IAS

IrLMP

IrLAP

Framer/Driver

Physical Layer

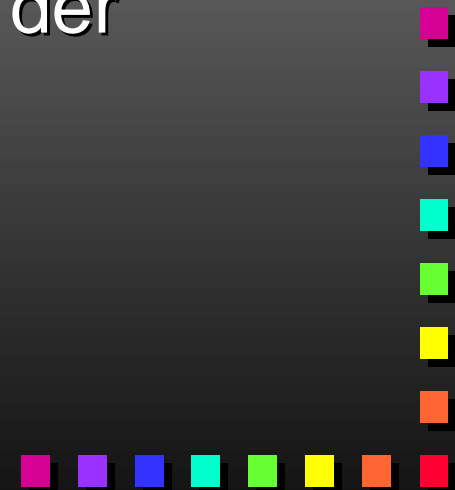
Information Access Service
Ankündigung und Zugang zu Diensten

flow control, segmentation & reassembly
multiplexing
device discovery, negotiation
bundle data packets/init hw, transceiver
nur halb-duplex, SIR, MIR, FIR, VFIR



IrDA / IAS

- IAS definiert:
 - **IAP**: Operationen zwischen IAS-Client und IAS-Server
 - Verhalten des IAS-Servers (z.B. über Zustands-Diagramme der beteiligten Geräte)
 - Informations-Modell

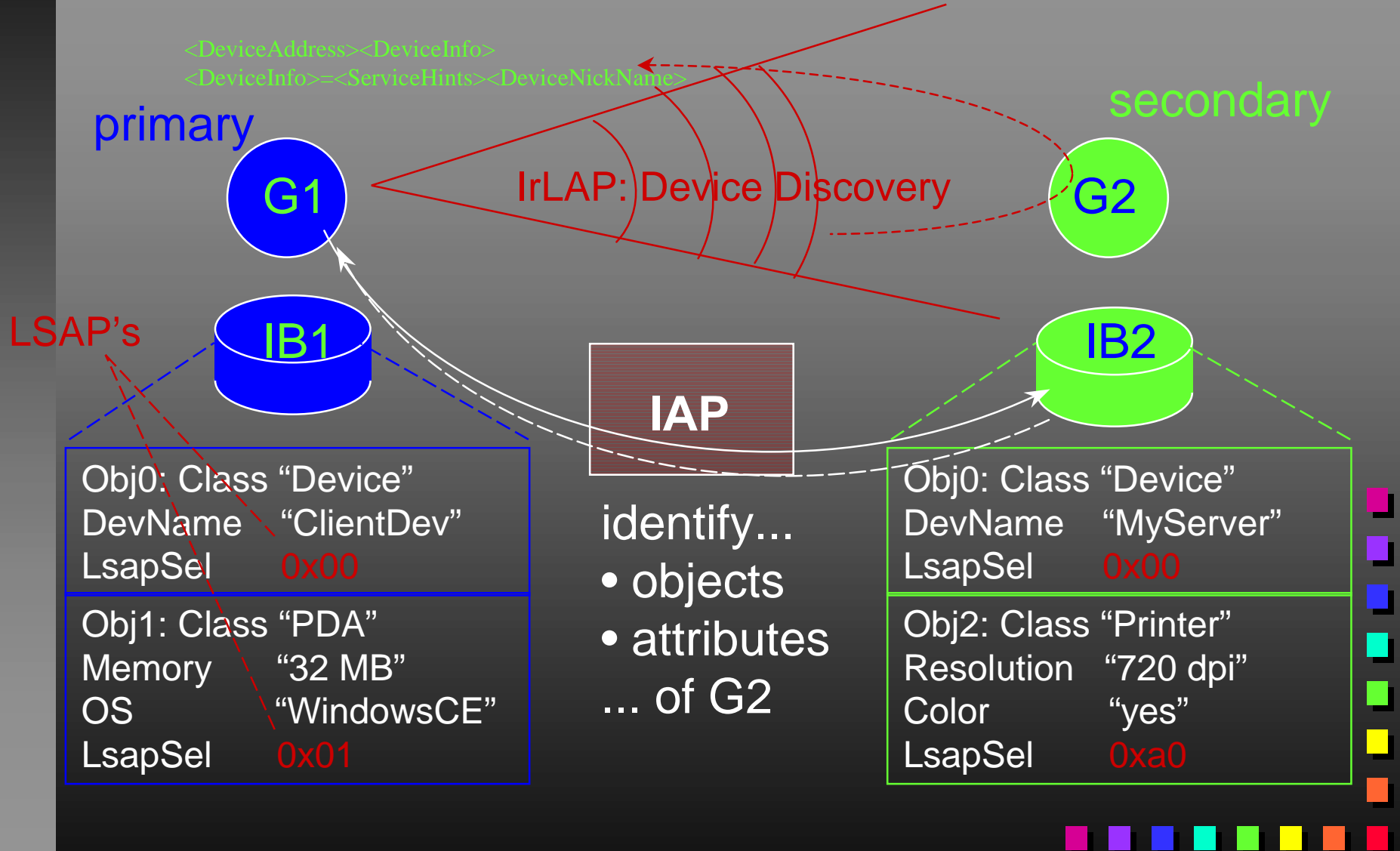


IrDA / IAS

- Jedes IrDA-Gerät besitzt einen IAS
- Registrieren eines Dienstes:
 - Applikation, die ein Dienst anbietet erhält einen *LSAP* (Link Service Access Point)
 - Dieser “Selektor” wird in der lokalen *information base* des Gerätes abgespeichert
 - Um auf die information base eines entfernten Gerätes zuzugreifen: *IAP*



Service-Primitiven & Informations-Modell



IrDA / IAS

Service-Suchmechanismus

- Über *Service-Hints* kann man erfahren, welche Art von Services angeboten werden (... grobe Geräte-Gliederung)
- Man weiss genau, welche Klasse man sucht und welche Attribute und Semantik sie hat



IrDA / IAS

Service-Suchmechanismus - Ein Beispiel

```
object 1 class
  FaxModemData{
  attribute
  IrDA:TinyTP:LsapSel =
    Integer(0x05);
}
```

```
/* Device Discovery */
DiscoveryList *dl= (slots);

/* Search Hints for Fax Device */
while (dl != NULL) {
  if( ) {
    DeviceAddress *da =dl->deviceAddress;
    break;
  }
  dl = dl->next;
}

/* Read the LMSAP Selector */
IASValue *lsapSel = (
  da, "FaxModemData",
  "IrDa:TinyTP:LsapSel");

/* Connect via lsapSel */ ...
```



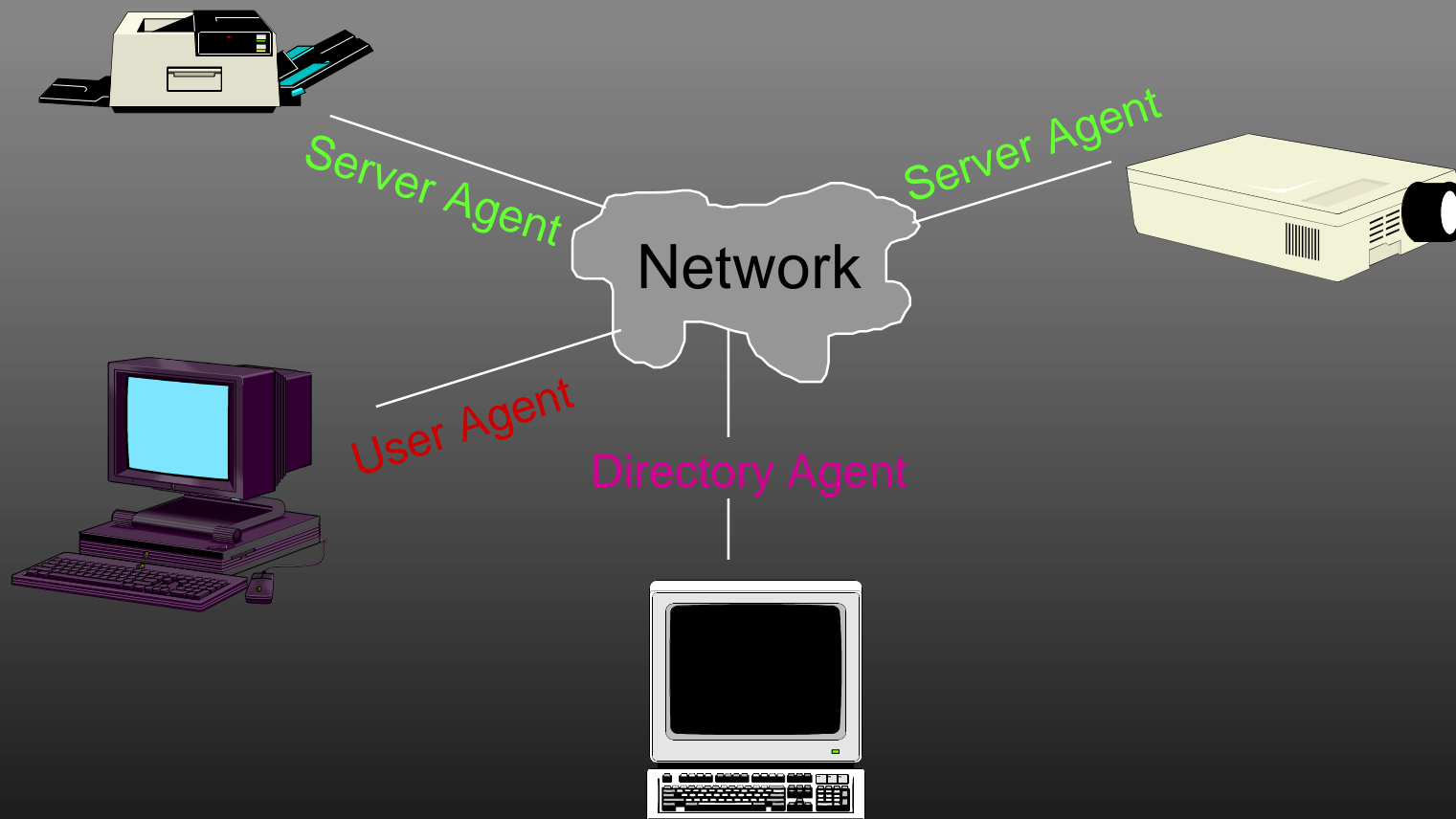
Uebersicht

- Einleitung & Motivation
- Service-Lokalisierung und Beschreibung in...
 - Bluetooth (SDP)
 - IrDA (IAS, IAP)
 - **SLP**
 - Jini
 - UPnP
 - Salutation
- Vergleich



SLP

IETF Vorschlag



SLP

- Mit Direcotry Agents (DAs)
 - Aktives finden von DAs
 - Passives finden von DA's
- Ohne DAs:
 - UAs stellen Service-Anfragen (multicast)
 - SAs antworten; dann unicast
 - Einfacher
 - Mehr Bandbreite

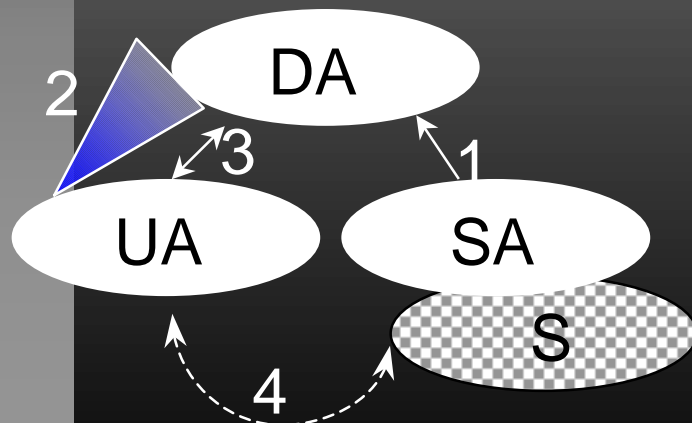


SLP

Service-Suchmechanismus

■ Service Templates

- Definieren Attribute von Dienst-Klassen
 - Deren default-Werte und Interpretation
- Von SAs gebraucht um Dienst zu registrieren
- Von UAs um Dienste zu finden

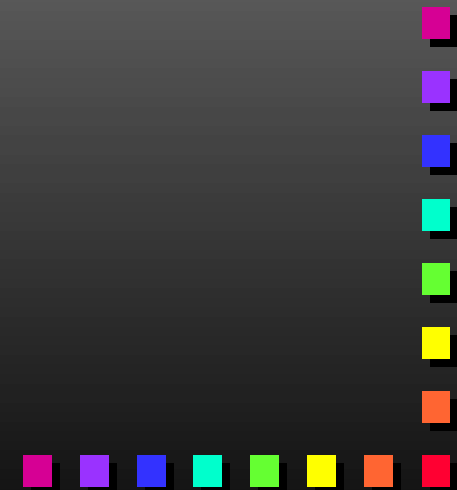


- 1 Service Register (URL, IPAddr, port, path)
- 2 DA Discovery
- 3 Anfrage (↗), URL (↖)
- 4 Service Nutzung



SLP

- Dienstbeschreibung
 - Typname (String)
 - URL
 - Menge von Schlüssel-Wert-Paaren
- Anfrage
 - Prädikat
- Service-Antwort

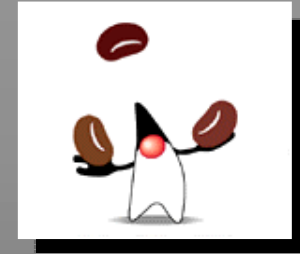


Uebersicht

- Einleitung & Motivation
- Service-Lokalisierung und Beschreibung in...
 - Bluetooth (SDP)
 - IrDA (IAS, IAP)
 - SLP
 - **Jini**
 - UPnP
 - Salutation
- Vergleich



Jini



■ Dienst

- Als Java-Objekt auf einem Server
- Spezifiziert durch Java-Interfaces

■ *Lookup-Service (JLS)*

- Zentrale Registrierungs- u. Vermittlungs-Instanz

■ Klient

- Versucht durch Lookup-Service zu einem *Dienst-Proxy-Objekt* zu kommen
- Kennt nur Interfaces, nicht Implement. (CWA)

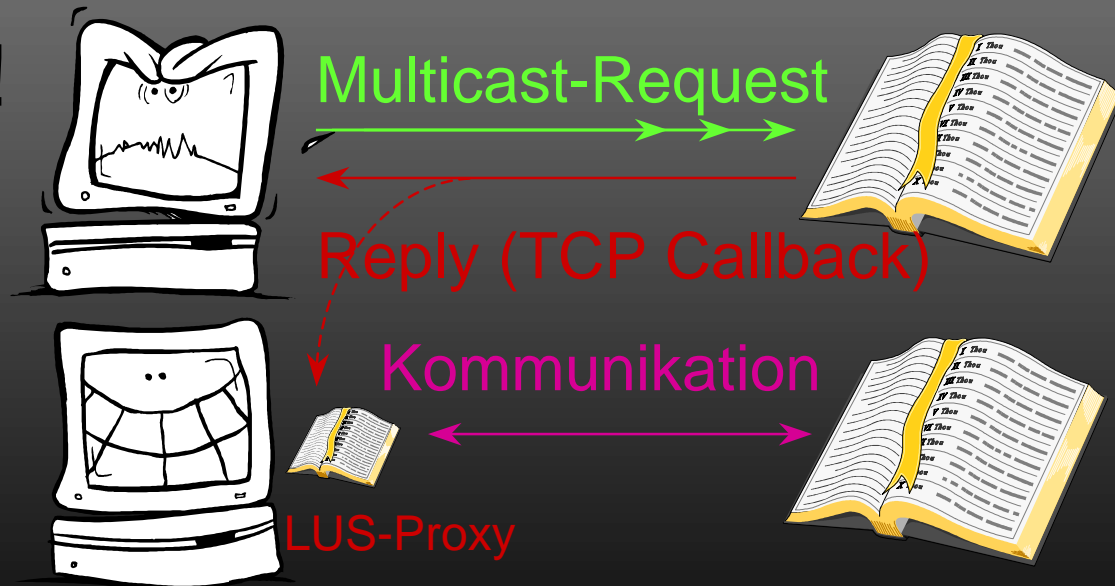


Jini

- Wie bekommen Dienst-Anbieter bzw. Dienst-Nutzer Kenntnis von den lokalen Gegebenheiten?

→ Discovery!

LUS-Proxy:
• register()
• lookup()
• notify()
• ...

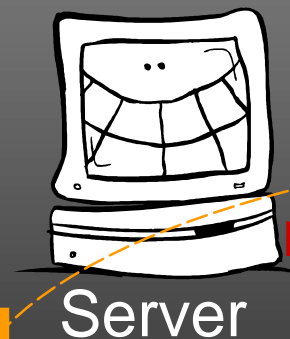


Jini

- Wie kann ein Server seinen Dienst anmelden?

→ Join!

Service-Item-Objekt:



LUS-Proxy

②

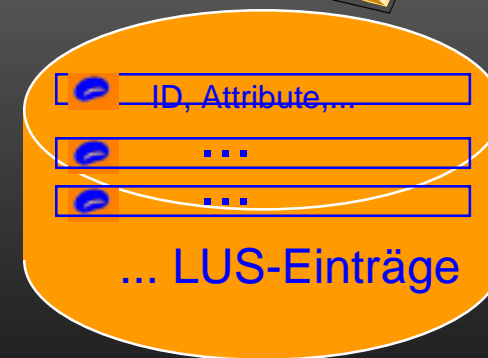
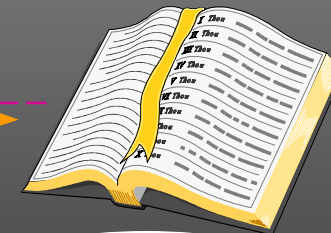
lease

register()

①

Service-Item-Objekt:

```
class ServiceItem {  
    ServiceID serviceID;  
    Object    service;  
    Entry[]  attributes;  
}
```



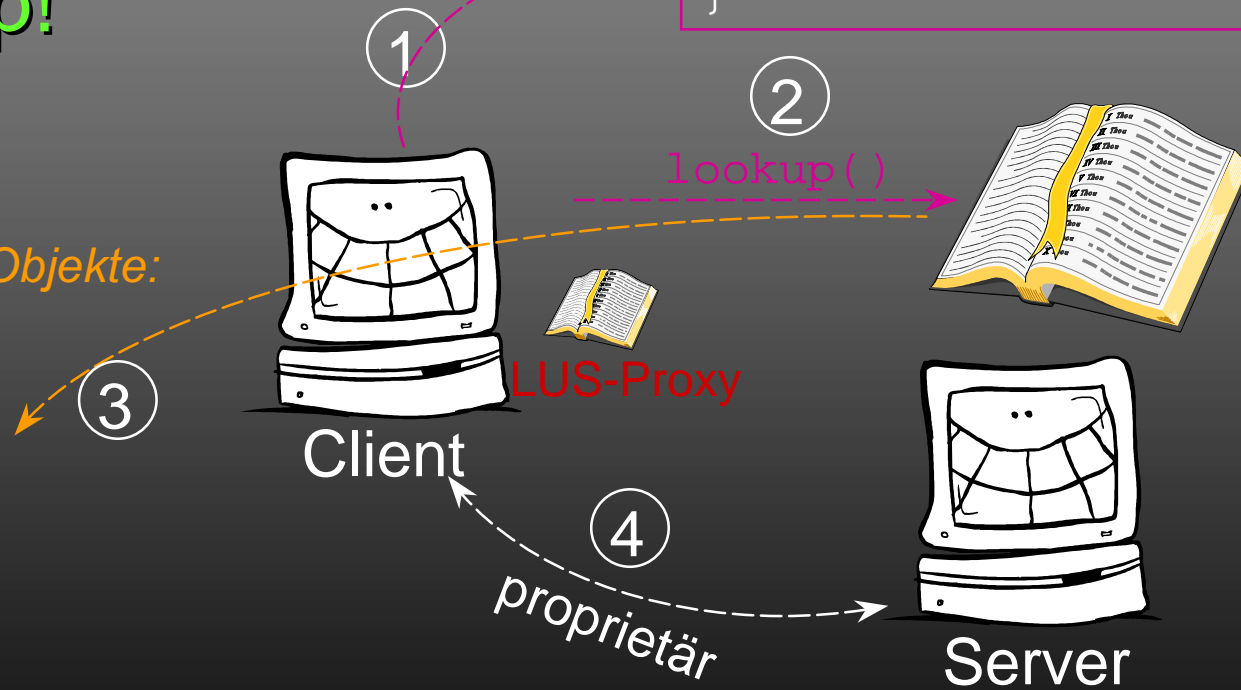
Jini

- Suchen eines Dienstes:
→ Lookup!

Service-Template-Objekt:

```
class ServiceTemplate {  
    ServiceID serviceID;  
    Class[]  serviceTypes;  
    Entry[]  attributes;  
}
```

Liste von
Service-Item-Objekte:



Jini

- Beschreibung: Java-Interfaces

- Entries als zusätzliche "Templates"
- `null`: wildcard-Semantik

```
class entry1
  implements Entry {
    String name;
  }
```

- Dienste: Methodenaufrufe (OO, RPC)

Anfrage:

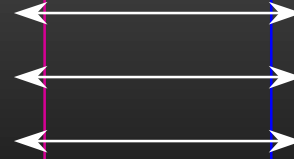
Service-Template-Objekt:

```
class ServiceTemplate {
  ServiceID serviceID;
  Class[]  serviceTypes;
  Entry[]  attributes;
}
```

In LUS gespeichert:

Service-Item-Objekte:

```
class ServiceItem {
  ServiceID serviceID;
  Object    service;
  Entry[]   attributes;
}
```



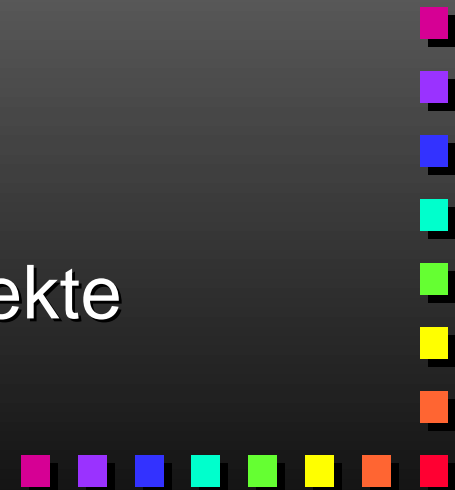
Uebersicht

- Einleitung & Motivation
- Service-Lokalisierung und Beschreibung in...
 - Bluetooth (SDP)
 - IrDA (IAS, IAP)
 - SLP
 - Jini
 - **UPnP**
 - Salutation
- Vergleich



UPnp

- Umfeld “Home/Office Network”
- “PnP” und Fernsteuerung dieser Geräte
- XML als Austauschformat von Geräte- und Dienstbeschreibung
- Spezifikation ist noch jung und anpassungsbedürftig
 - z.B. noch keine Sicherheits-Aspekte



UPnP / SSDP

■ Dienstvermittlung

- Mit oder ohne zentrale Komponente, *SSDP-Proxy*, möglich

■ Dienst-Anfrage

- Klient sendet Multicast auf definiertem Kanal
- Enthält Dienst-Typ-Information (String)
- Server/Proxy, die gewünschten Dienst anbieten/vermitteln: UDP-Paket zurücksenden



UPnP / SSDP

- UDP-Nachricht enthält u. A. Description-URL
 - SSDP-Klient kann dadurch
 - XML-Beschreibung des Dienstes einsehen (def. DTD)
 - Modell, SN, Herstellerangaben, ...
 - Kontroll-Kommandos an Dienst senden
 - SOAP: Simple object access protocol, "RPC auf XML"
- GENA Generic Event Notification Architecture



Uebersicht

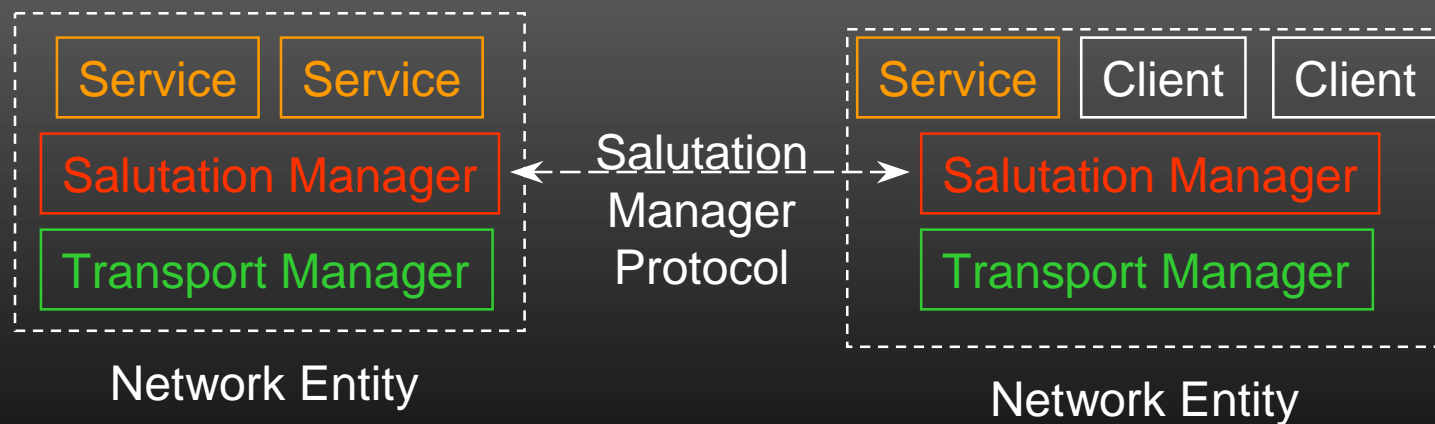
- Einleitung & Motivation
- Service-Lokalisierung und Beschreibung in...
 - Bluetooth (SDP)
 - IrDA (IAS, IAP)
 - SLP
 - Jini
 - UPnP
 - **Salutation**
- Vergleich



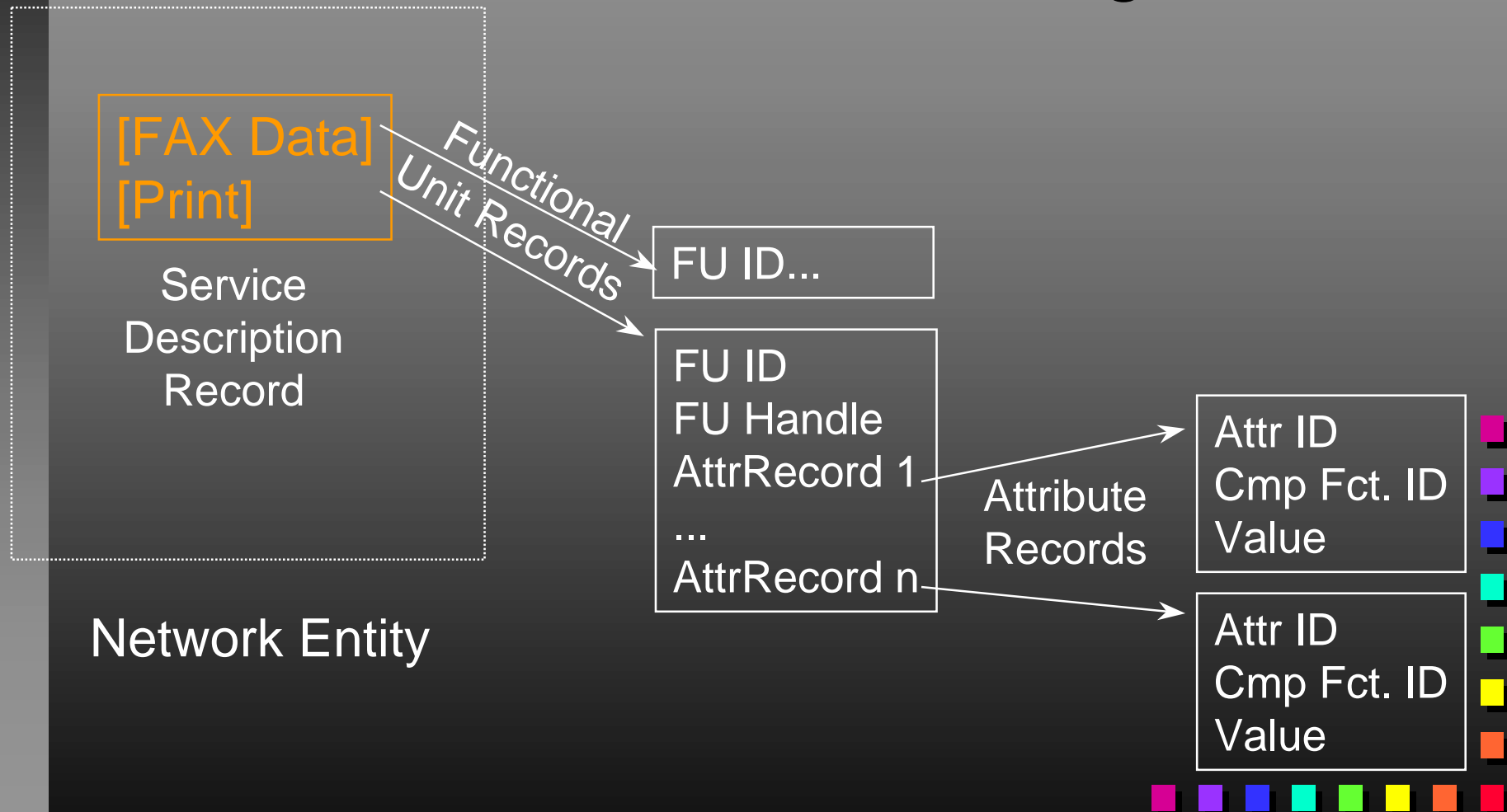
Salutation

■ Einheiten:

- **Functional Units:** Wichtige Service-Eigenschaften
- **Salutation Manager:** Service Broker
- **Transport Manager:** Netzwerk-Unabhängigkeit

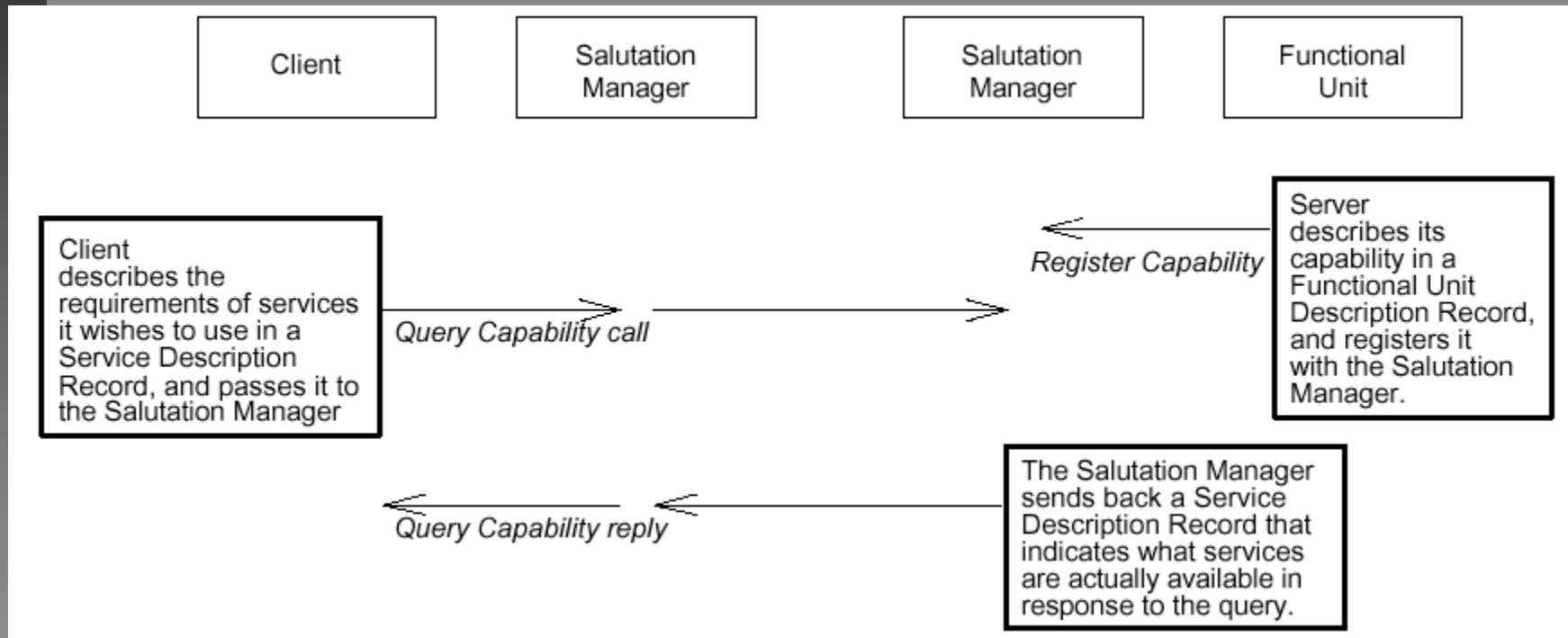


Salutation: Dienst-Beschreibung



Network Entity

Salutation



Uebersicht

- Einleitung & Motivation
- Service-Lokalisierung und Beschreibung in...
 - Bluetooth (SDP)
 - IrDA (IAS, IAP)
 - SLP
 - Jini
 - UPnP
 - Salutation
- **Vergleich**



Eigenschaften

Kriterien	Bluetooth	IrDA	SLP	Jini	UPnP	Salutation
Service-Registrierung	Nur lokal	Nur lokal	Authent. Multicast	Multicast m. TCP-Callback/RMI	Multicast/Unicast	Nur lokal
Service-	UUID	String	String	Java-Interface	String	String
Service Beschreibung	Attribute/Werte	Attribute/Werte	Attribute/Werte	Serialisierte Java-Objekte	String u. XML-Instanz	Attribute/Werte
Klienten-Anfrage	UUID	String	Boolsche Ausdrücke	Java-Objekt-Template	String	String
Dezentraler Betrieb	Ausschliesslich	Ausschliesslich	Möglich	(nein)	Möglich	Ausschliesslich



Vergleich

Kriterien	Bluetooth	IrDA	SLP	Jini	UPnP	Salutation
Service Discovery	✓	✓	✓	✓	✓	✓
Service Announcement	✗	✗	✓	✓	✓	✓
Interoperability	✓	✓	✓	✓	✗	✓
Security	✓	✓	✓	✓	✗	✗



Vergleich

■ Bluetooth

- + Effiziente Suchmöglichkeiten
- + “Navigations”-Möglichkeiten
- (±) Omnidirektional
- Nur Suche nach UUID
- Nur Information über Service
- (–) Teuer (im Moment)
- Limitiert auf 1 MB/s

■ IrDA

- + Billig, Verbreitet u. Einfach zum Implementieren
- + Strings
- Limitiert auf Distanz: 1m
- (±) Geräte müssen positioniert werden



Vergleich

■ SLP

- + Flexibel: Mit/Ohne DA
- + Gut skalierbar

– Nur für TCP/IP-Netzwerke

■ Jini

- + Mächtig
- + Plattform-Unabhängig (?)

– Setzt auf jedem Gerät
JVM voraus

– Nur für TCP/IP-Netzwerke



Vergleich

■ UPnP

+ XML für Geräte-Protokoll-Standardisierung

- Discovery nur auf Subnet
- Nur für TCP/IP-Netzwerke

■ Salutation

+ Netzwerk-Unabhängig
+ Kollaboration von SM untereinander

- Spezifikation enthält keinerlei Sicherheits-Aspekte



Abschliessend

- Unterschiede
 - Zentral vs. Dezentral
 - Dienstbeschreibung
 - Objekte, Strings, Key/Value, XML-Dokumente,..
 - Sicherheit
- Heute
 - Interoperabilität erwünscht
- Morgen ???

